

Terms that we use in RDMBS:

Relation : Table : Collection of records

Column / Attribute /Field : Feature of an entity ( rn , name , address)

Row / Record / Tuple : Collection of data for a single entity

Every Relation should follow some Integrity Rules

Rule No 1: Entity Integrity Rule : Primary Key : Unique value for every record , id, rn

* No duplicacy
* Not Null

Rule No 2: Referential Integrity Rule

Foreign Key : column whose value could be value of primary key of some other table OR could be null ( We use it in case of relation between more than 1 table)

Rule No 3: Domain Integrity Rule : Columns can have some predefined values OR range of values

Pool of values from which a column can derive its value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RollNo | Name | Address | Marks | State | City |
| 1 | Ajay | Delhi | 67 | Delhi | N Delhi |
| 2 | Deepak | Delhi | 89 | Delhi | N Delhi |
| 3 | Shashi | Delhi | 90 | Calcutta |  |
| 4 | Ajay |  |  |  |  |

No of Records : Cardinality of a Relation : 4

Unary Table : Only one column

Binary Table : Table with 2 columns

Ternary

4-nary

Degree of a Relation : No, of Columns : 6 , 6nary

Fees

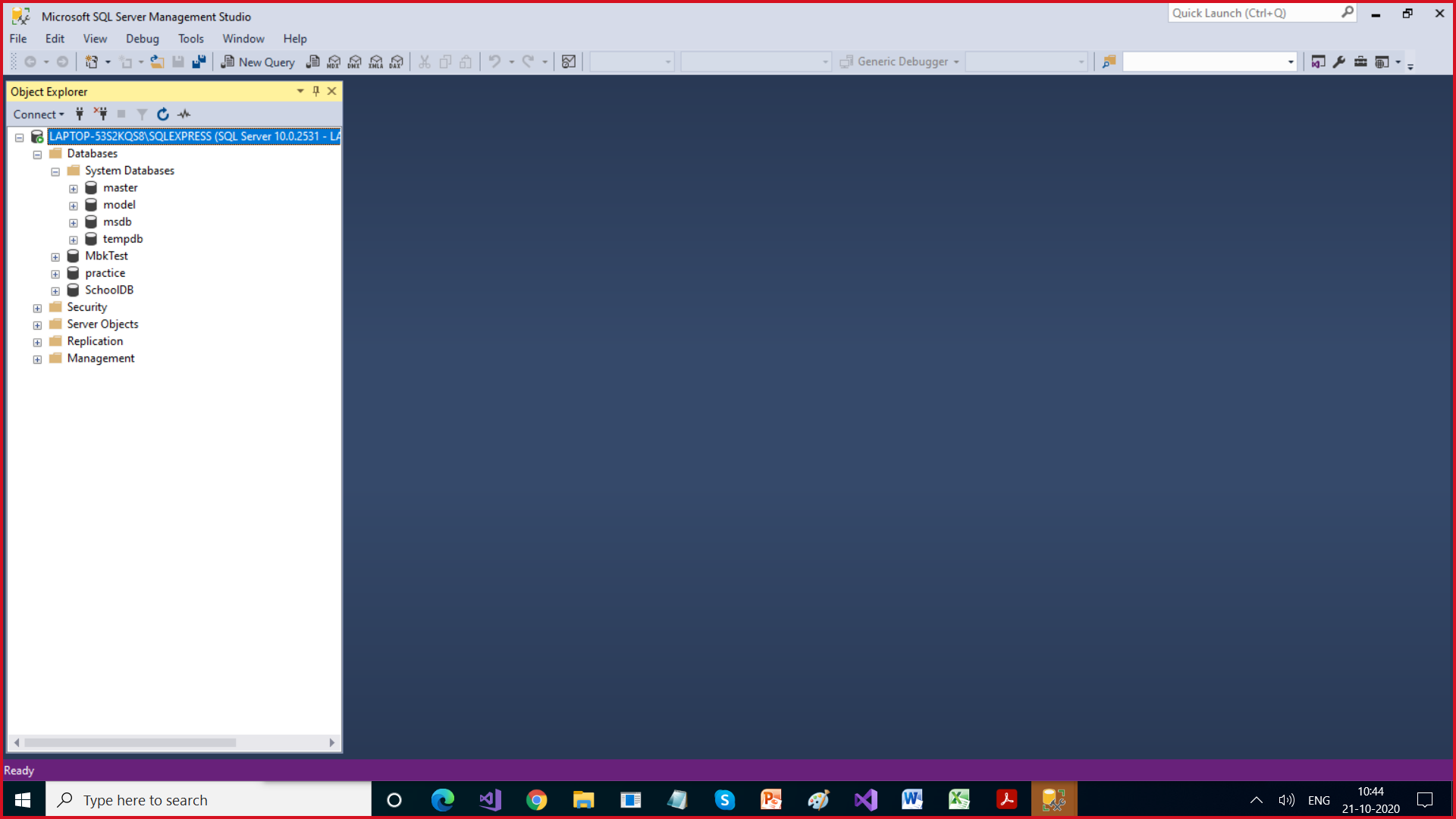
|  |  |
| --- | --- |
| RollNo | FeesPaid |
| 3 | 10000 |

No of Records : Cardinality of a Relation : 1

Degree of a Relation : No, of Columns : 2 , Binary

Database: Collection of Records

Sql Server : Database   
: Collection of objects (like table, stored procedure, functions, view . etc…)



<https://alwayssql.wordpress.com/2018/02/23/sql-server-system-databases/#:~:text=System%20Databases%201%20Master%20Database.%20Master%20is%20the,Agent%20information%2C%20SQL%20Server%20jobs.%20More%20items...%20>

SQL : Structured Query Language :

This is a language used to work with RDBMS

Easy to learn , syntax is same as english language

Case Insensitive

Hello HELLO

**SQL Statements**

DDL : Data Definition Language , create , alter , drop , truncate table (schema)

DQL : Data Query Language : Select

DML : Data Manipulation Language : Select, Insert , Delete , Update (Records)

TCL : Transaction Control Language : Commit , RollBack

DCL : Data Control Language : Grant , Revoke

-- Create Database

create database EmployeeDB

use EmployeeDB

create table Employee (

id int,

name varchar(20),

address varchar(20),

salary int

)

insert into Employee values(1,'Ajay','Delhi',12000)

insert into Employee values(2,'Vijay','Delhi',21000)

insert into Employee values(3,'Sagar','Calcutta',18000)

insert into Employee values(4,'Prateek','Delhi',52000)

insert into Employee values(5,'Ravi','Calcutta',9000)

insert into Employee values(6,'Ajay Singh','Gurgaon',34000)

insert into Employee values(7,'Deepak','Gurgaon',45000)

Select \* from Employee

--Projection

Select id, name from Employee

-- Selection

Select \* from Employee

where id = 9

Select \* from Employee

where id < 9

Select \* from Employee

where id < 3

Select \* from Employee

where address = 'Delhi'

Operators

Arithmatical : + - / \* %

Relational/Comparison : = <> > < >= <=

Logical : AND OR NOT

Between : Range

IN : Range

LIKE : Pattern Matching

% : Any no. of charcaters

\_ : 1 character

--DDL

-- Create Database

create database EmployeeDB

use EmployeeDB

--DDL

create table Employee (

id int,

name varchar(20),

address varchar(20),

salary int

)

--DML

insert into Employee values(1,'Ajay','Delhi',12000)

insert into Employee values(2,'Vijay','Delhi',21000)

insert into Employee values(3,'Sagar','Calcutta',18000)

insert into Employee values(4,'Prateek','Delhi',52000)

insert into Employee values(5,'Ravi','Calcutta',9000)

insert into Employee values(6,'Ajay Singh','Gurgaon',34000)

insert into Employee values(8,'Deepak','Delhi',45000)

--DML / DQL

Select \* from Employee

--Projection

Select id, name from Employee

-- Selection

Select \* from Employee

where id = 9

Select \* from Employee

where id < 9

Select \* from Employee

where id < 3

Select \* from Employee

where address = 'Delhi'

Select \* from Employee

select \* from Employee where salary >= 12000 and salary <= 20000

Select \* from Employee where salary between 12000 and 20000

select \* from Employee where address = 'Delhi' OR address ='Gurgaon'

select \* from Employee where address in ('Delhi','Gurgaon')

Select \* from Employee where name LIKE '%g%'

select \* from Employee order by name desc

select \* from Employee order by salary , address

--DDL

-- alter table is used for changing structure of table

alter table Employee add department varchar(20)

--DML/DQL

select \* from Employee

--DML

-- update is used to edit record or change record

update employee set department ='HR'

update employee set department ='Accts'

where id between 4 and 7

update employee set department ='HR'

where id = 8

-- Change width of address to 30

--DDL

alter table Employee alter column address varchar(30)

--DDL

-- Delete column of a table

alter table Employee drop column address

-- Copy one table to other , creates new table

select \* into Employee1 from Employee

select \* from Employee1

--DML

--Delete Records

delete Employee where id=8

delete Employee

Select \* from Employee

--DDL

Drop table Employee

select \* into Employee from Employee1

DDL Command are self committed

DML commands are not self committed

Delete Employee

Select \* from Employee

rollback

--DDL

--Delete all records

truncate table Employee

Delete is DML

Truncate Table is DDL

Delete is not self committed , we can use Rollback to retrieve deleted records

Truncate Table is self committed , we can not use Rollback to retrieve deleted records

With Delete , we can add conditions

With Truncate Table , we can not use where clause

--DML

insert into Employee(id,name,address,salary) values(1,'Ajay','Delhi',12000)

insert into Employee(id,address,salary,name) values(1,'Delhi',22000,'Deepak')

-- Third Use

insert into Employee(id,salary,name) values(10,22000,'Deepak')

Select \* from Employee

select id, name from Employee

Select id "Employee ID" , name as "Employee Name" from Employee

Select id , name , salary , salary + 2000 from EMployee

Select id , name , salary , salary + 2000 as "Updated Salary" from EMployee

select sum(salary) as "Total Salary" , max(salary) as "Max Salary" , min(salary), avg(salary) from Employee

select sum(salary) as "Total Salary" , max(salary) as "Max Salary" , min(salary), avg(salary) from Employee

alter table Employee add age int

select \* from Employee

update Employee set age = 26

select upper(name) as "Name of Employee" from Employee

select lower(name) as "Name of Employee" from Employee

select left(name,4) as "First 4 characters of Name" from Employee

select substring(name,1,4) as "First 4 characters of Name" from Employee

select name, charindex('a',name) from Employee

select \* from Employee where id in (101,103,105)

select \* from Employee where id between 101 and 105

select \* from Employee where department in ('HR','Accts')

Select \* from Employee where name LIKE 'A%'

select \* from Employee where charindex('a',name) >0

select \* from Employee where department NOt IN ('HR','Accts')

and salary not between 1000 and 2000

Constraints

drop table Employee

create table Employee

(

--id int primary key

id int constraint pk primary key,

name varchar(20) not null constraint name\_length check(len(name)>20),

salary int constraint salary\_constraint check(salary between 10000 and 25000),

department varchar(20) constraint dept\_constraint check(department in ('HR','Accts')),

manager varchar(20) unique

)

alter table EMployee add marks int default 40 WITH VALUES

alter table EMployee alter column name varchar(30)

select \* from Employee

insert into Employee (id,name, salary, department,manager)

values(1,'Ajay Kumar Singh1212',12000,'HR','Ajay')

insert into Employee (id,name, salary, department,manager)

values(2,'Ajay Kumar Singh1212',12000,'HR','Vijay')

insert into Employee (id,name, salary, department,manager)

values(3,'Ajay Kumar Singh1212',12000,'HR',null)

insert into Employee (id,name, salary, department,manager)

values(5,'Ajay Kumar Singh1212',12000,'Accts','Sandeep')

-- id int primary key,

drop table Employee

create table Employee

(

--id int primary key

id int constraint pk primary key,

name varchar(20) not null constraint name\_length check(len(name)>20),

salary int constraint salary\_constraint check(salary between 10000 and 25000),

department varchar(20) constraint dept\_constraint check(department in ('HR','Accts')),

manager varchar(20) unique

)

alter table EMployee add marks int default 40 WITH VALUES

alter table EMployee alter column name varchar(30)

select \* from Employee

insert into Employee (id,name, salary, department,manager)

values(1,'Ajay Kumar Singh1212',12000,'HR','Ajay')

insert into Employee (id,name, salary, department,manager)

values(2,'Ajay Kumar Singh1212',12000,'HR','Vijay')

insert into Employee (id,name, salary, department,manager)

values(3,'Ajay Kumar Singh1212',12000,'HR',null)

insert into Employee (id,name, salary, department,manager)

values(5,'Ajay Kumar Singh1212',12000,'Accts','Sandeep')

-- id int primary key,

-- Remove constraint from table

alter table Employee drop constraint dept\_constraint

-- alter table Employee alter constraint name\_length

2nd Nov

Use model -- 5 tables 2 sp

create table a(id int)

create database S2 -- 5 tables 2 sp

SQl is language case insensitive

use s2

select \* from sys.tables

-- SQL -- comments Lines ignored For documentation purpose

create database Employee -- DDL

-- use database

use Employee

-- DDL

create table Emp (

id int,

name varchar(20),

department varchar(20),

salary int)

-- DML

insert into Emp values(1,'Ajay','HR',12000)

insert into Emp values(2,'Deepak','Accts',23000)

insert into Emp values(3,'Vijay','HR',12000)

insert into Emp values(4,'Sagar','Sales',21000)

-- implicit way of storing null in a column

insert into Emp(name,department,salary) values('Sagar','Sales',21000)

-- explicit way of storing null in a column

insert into Emp values(null, null,'Sales',null)

-- DML / DQL

select \* from Emp

-- \* means all columns

-- Projection / Vertical Subset

-- Display some columns

Select id EmployeeID , name EmployeeName from Emp

Select id as EmployeeID , name as EmployeeName from Emp

Select id as "Employee ID" , name as "Employee Name" from Emp

-- Selection / Horizontal SubSet

-- where clause

Select \* from Emp where id = 3

-- Operator

-- id and 3 are operands

-- = opertaor

Select \* from Emp where id > 3

Select \* from Emp where name ='Ajay'

Select \* from Emp where salary > 10000 and salary < 15000

-- Between , Range operator , include the boundaries also

Select \* from Emp where salary between 10000 and 15000

Select \* from Emp where department= 'HR' OR department='Accts'

-- IN Operator

Select \* from Emp where department IN('HR','Accts', 'Sales')

-- To check for null , we use is operator

Select \* from Emp where id is null

Select \* from Emp where id is not null

-- like is for pattern matching

-- % means any no. of character

-- \_ means one character

select \* from Emp where name like '%a%'

select \* from Emp where name like '%i%'

Select \* from Emp where name like 'V\_\_\_\_'

Select \* from Emp where name like 'V%'

Select \* from Emp where name like '%a'

Select id, name , salary from Emp

-- Calculated Column // Derived Column

Select id, name , salary , salary + 10000 from Emp

Select id, name , salary , salary + 10000 As "Incremented Salary" from Emp

-- insert multiple records in one command

insert into Emp values

(5,'Ajay','HR',12000),

(6,'Deepak','Accts',23000),

(7,'Vijay','HR',12000),

(8,'Sagar','Sales',21000)

--Want to add one column

alter table emp add address varchar(20)

select \* from Emp

alter table emp add city varchar(20) default '1' with values

select \* from Emp

-- update Records

update Emp set address ='Delhi'

update Emp set address ='Calcutta' where id between 1 and 3

-- Delete the Records

delete emp

delete from emp

delete from emp where id is null

select \* from Emp